

CLAIM AMENDMENTS

1 1. (Currently Amended) A method for automatically generating a description of a data
2 exchange format based on computer program source code expressed in a source language, the
3 method comprising the computer-implemented steps of:

4 receiving, from a source code file, comment data including first data indicating a
5 parameter of the data exchange format, wherein the comment data is ignored
6 by a source code processor of the source language;

7 receiving from the source code file second data, associated with the comment data,
8 indicating a statement that defines a class of data objects in the source
9 language; and

10 automatically generating, based on the first data and the second data, third data that
11 describes the data exchange format, wherein the third data comprises
12 instructions defining a mapping between data objects of the class of data
13 objects and data items having the data exchange format.

1 2. (Original) A method as recited in Claim 1, further comprising generating, based on the
2 first data and the second data, a module to convert a data object of the class of data objects
3 into a data item of the data exchange format as described by the third data.

1 3. (Original) A method as recited in Claim 1, further comprising generating, based on the
2 third data, a module to convert a data object of the class of data objects into a data item of the
3 data exchange format as described by the third data.

1 4. (Currently Amended) A method as recited in Claim 1, further comprising generating,
2 based on the first data and the second data, a module to derive a data object of the class of
3 data objects from a data item of the data exchange format as described by the ~~second~~third
4 data.

1 5. (Original) A method as recited in Claim 1, further comprising generating, based on the
2 third data, a module to derive a data object of the class of data objects from a data item of the
3 data exchange format as described by the third data.

1 6. (Original) A method as recited in Claim 1, wherein the third data is formatted according
2 to a database query language.

1 7. (Original) A method as recited in Claim 1, wherein the third data is formatted according
2 to a symbolic markup language.

1 8. (Original) A method as recited in Claim 1, wherein the third data is formatted according
2 to extensible markup language (XML).

1 9. (Original) A method as recited in Claim 1, wherein the third data comprises one or more
2 statements in an XML schema document.

1 10. (Original) A method as recited in Claim 1, wherein the third data is one or more
2 statements in an XML document type definition (DTD) document.

1 11. (Original) A method as recited in Claim 1, wherein the third data is one or more
2 statements in an XML document type definition (DTD) document, and wherein the
3 parameter is at least one of a root element associated with an entire DTD document, an
4 element and an attribute of an element.

1 12. (Original) A method as recited in Claim 1, wherein the third data is one or more
2 statements in an XML document type definition (DTD) document, and wherein the first data
3 includes one or more properties of the parameter.

1 13. (Currently Amended) A method as recited in Claim 1, wherein the source language is
2 ~~Java~~@JAVA.

1 14. (Currently Amended) A method as recited in Claim 1, wherein the source language is
2 ~~Java~~@JAVA, and wherein the first data includes a tag for an automated ~~Java~~JAVA
3 documentation system.

1 15. (Currently Amended) A method as recited in Claim 1, wherein the source language is
2 ~~Java~~@JAVA, wherein the first data includes a tag for an automated ~~Java~~JAVA
3 documentation system, and wherein the tag is a user-defined tag for the ~~Java~~JAVA
4 documentation system.

1 16. (Currently Amended) A method as recited in Claim 1, wherein the source language is
2 ~~Java~~@JAVA, wherein the first data includes a tag for an automated ~~Java~~JAVA
3 documentation system, wherein the tag is a user-defined tag for the ~~Java~~JAVA
4 documentation system, and wherein said step of generating the third data is performed by a
5 user-defined routine invoked by the automated ~~Java~~JAVA documentation system in response
6 to the tag.

1 17. (Currently Amended) A method for binding a data exchange format with an
2 application having source code in a particular language, the method comprising the steps of:
3 inserting, into the source code within comment data ignored according to a processor
4 for the particular language, first data including a tag and a parameter of a data
5 exchange format; and
6 causing a processor to produce second data for configuring the data exchange format
7 based at least in part on the first data, wherein the second data comprises
8 instructions defining a mapping between data objects and data items in the
9 data exchange format.

1 18. (Currently Amended) A method as recited in Claim 17, further comprising the steps
2 of:
3 causing the processor to produce at least one of a module for marshaling data objects
4 into a data item in the data exchange format as configured by the second data
5 and a module for de-marshaling data objects from a data item in the data
6 exchange format as configured by the second data; and
7 building an executable version of the application based on the source code and at least
8 one of the module for marshaling data and the module for de-marshaling data.

1 19. (Currently Amended) A method as recited in Claim 18, wherein the particular
2 language is the JavaJAVA language, wherein the tag is a user-defined tag of an automated
3 JavaJAVA documentation system; and wherein said step of causing a processor to produce
4 the second data further comprises providing a routine, invoked by the automated JavaJAVA
5 documentation system in response to the tag, to produce the second data.

1 20. (Currently Amended) A method as recited in Claim 19, wherein the particular
2 language is the JavaJAVA language; wherein the tag is a user-defined tag of an automated
3 JavaJAVA documentation system; and wherein said step of causing a processor to produce at
4 least one of the module for marshaling and the module for de-marshaling further comprises
5 providing a routine, invoked by the automated JavaJAVA documentation system in response
6 to the tag, to produce at least one of a JavaJAVA module for marshaling and a JavaJAVA
7 module for de-marshaling.

1 21. (Currently Amended) A computer-readable medium carrying one or more sequences
2 of instructions for binding a data exchange format with an application having source code in
3 a particular language, which instructions, when executed by one or more processors, cause
4 the one or more processors to carry out the steps of:
5 receiving, from a particular file that includes the source code, comment data including
6 first data indicating a parameter of the data exchange format, wherein the

comment data is ignored by a source code processor of the particular language;
receiving from the particular file second data, associated with the comment data, indicating a statement that defines a class of data objects in the particular language; and
generating, based on the first data and second data, third data for configuring the data exchange format, wherein the third data comprises instructions defining a mapping between data objects of the class of data objects and data items having the data exchange format.

22. (Currently Amended) An apparatus for binding a data exchange format with an application having source code in a particular language, comprising:

means for receiving, from a particular file that includes the source code, comment data including first data indicating a parameter of the data exchange format, wherein the comment data is ignored by a source code processor of the particular language;
means for receiving from the particular file second data, associated with the comment data, indicating a statement that defines a class of data objects in the particular language; and
means for generating, based on the first data and second data, third data for configuring the data exchange format, wherein the third data comprises instructions defining a mapping between data objects of the class of data objects and data items having the data exchange format.

23. (Currently Amended) An apparatus for binding a data exchange format with an application having source code in a particular language, comprising:

a processor;
one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:
receiving, from a particular file that includes the source code, comment data including first data indicating a parameter of a data exchange format,

8 wherein the comment data is ignored by a source code processor of the
9 particular language;
10 receiving from the particular file second data, associated with the comment
11 data, indicating a statement that defines a class of data objects in the
12 particular language; and
13 generating, based on the first data and second data, third data for configuring
14 the data exchange format, wherein the third data comprises
15 instructions defining a mapping between data objects of the class of
16 data objects and data items having the data exchange format.